Handling New Class in Online Label Shift

Yu-Yang Qian, Yong Bai, Zhen-Yu Zhang, Peng Zhao, *Member, IEEE*, and Zhi-Hua Zhou, *Fellow, IEEE*

***Abstract*—In many real-world applications, data are continuously accumulated in open environments, and new classes may emerge over time. For instance, in disease diagnosis, the prevalence of a certain disease may vary seasonally, and new diseases can also emerge. This paper investigates the problem of learning from unlabeled data stream where the *label distribution evolves over time*, and meanwhile, *previously unseen new classes may appear*. To handle the emerging new classes in online label shift, we first design a novel risk estimator by unbiased risk rewriting and mixture proportion estimation, which enables the identification of new class data. Subsequently, we employ the online ensemble paradigm for model updating to handle unknown distribution shifts. Moreover, we introduce the sketching and ensemble pruning mechanisms to improve the efficiency of the algorithm, making it more lightweight and practical. The proposed approach enjoys a theoretical guarantee of dynamic regret, ensuring its effectiveness in adapting to the unknown distribution shifts and the emergence of new classes in streaming data. Experiments on diverse benchmark datasets and two real-world applications demonstrate the effectiveness of the algorithm.**

***Index Terms*—Data stream, distribution shift, new class, online label shift, weakly supervised learning.**

I. Introduction

MACHINE learning algorithms have made significant successes across various applications [1], typically relying on the assumption that the training and testing data are generated from an identical distribution. However, in many real-world tasks, the testing data are continuously collected from open environments, resulting in a distribution mismatch between the training and testing data, and the distribution of testing data can even change over time [2], [3]. Furthermore, owing to the streaming nature of data, *new class data could appear*, presenting instances that were not encountered previously. Therefore, it is essential to adaptively learn from unlabeled data streams with changing distributions, particularly with the emergence of new classes.

In this paper, we investigate the problem of handling new classes in online label shift. Specifically, the learner can have some offline labeled data for model training. However, during online testing phase, *unlabeled data* continuously arrives with its *label distribution changing over time* [4]; simultaneously, *new class data could appear* in online unlabeled data stream, as shown in Fig. 1. The learner is required to continuously adapt to the changing distribution and accommodate the arrival of new classes. This problem is crucial because it encompasses various real-world tasks. For instance, considering disease diagnosis tasks, the prevalence of a certain disease may vary across seasons [5], which induces continuous label shifts. Moreover, the emergence of new diseases that were not encountered in the initial labeled data [6] can pose a significant challenge in handling these emerging new classes.

Existing approaches primarily focused on either handling online label shifts within a fixed label space, or dealing with new classes while employing a fixed classifier for known classes. As a typical kind of distribution change, online label shift, characterized by continuous changes in the label distribution of unlabeled data stream, has garnered substantial interest in the literature [4], [7], [8], [9], [10], [11]. This line of research first estimates the underlying loss of online unlabeled data in an unbiased manner, followed by formulating the problem as an online convex optimization problem. However, these studies do not consider the appearance of new classes in the open environments, which is a common occurrence in many real-world tasks. Another line of research on handling the new classes focuses on handling only new classes within unlabeled data stream [12], [13]. This line of research uses various anomaly detectors to detect new classes and updates models accordingly. However, these studies mainly concentrate on detecting new classes while disregarding distribution changes within the known class data, which may cause a degradation in the overall performance. Additionally, heuristic mechanisms for identifying new class data possess limited fitting capabilities and lack theoretical guarantees. It is noteworthy that in numerous real-world scenarios, the issue of label shifts and the new class can occur simultaneously, posing potential challenges to the existing algorithms.

In this paper, we initiate and investigate the problem of handling *New class in Online Label Shift* (N-OLS), which encompasses a wide range of real-world tasks. In particular in this scenario, more and more unseen new classes are allowed to successively emerge in the data stream as time evolves. Although previous works have studied the new class and label shift problems separately, the *conjunction of online label shift and new classes* presents new challenges, especially for unlabeled data streams. On one hand, the emergence of new classes can

Received 30 July 2024; revised 1 April 2025; accepted 10 June 2025. Date of publication 25 June 2025; date of current version 24 July 2025. This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0114800 and in part by NSFC under Grant U23A20382. The work of Zhen-Yu Zhang was supported by JSPS KAKENHI under Grant JP25K21282. Recommended for acceptance by R. Akbarinia. *(Corresponding author: Zhi-Hua Zhou.)*

Yu-Yang Qian, Yong Bai, Peng Zhao, and Zhi-Hua Zhou are with the National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, Nanjing 210093, China (e-mail: qianyy@lamda.nju.edu.cn; baiy@lamda.nju.edu.cn; zhaop@lamda.nju.edu.cn; zhouzh@lamda.nju.edu.cn).

Zhen-Yu Zhang is with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: zhen-yu.zhang@riken.jp).

Digital Object Identifier 10.1109/TKDE.2025.3583138

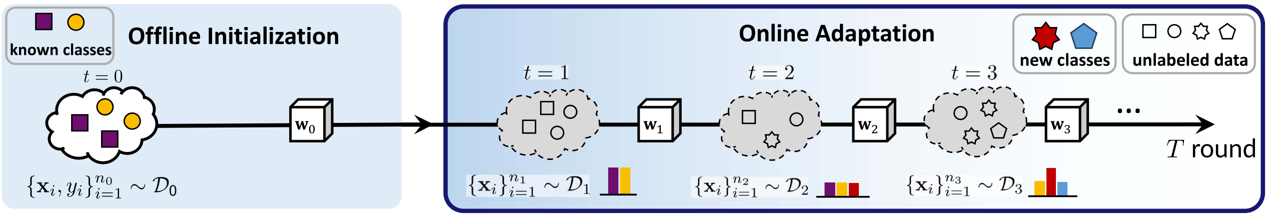

known classes
Offline Initialization
$t = 0$
$\mathbf{w}_0$
$\{\mathbf{x}_i, y_i\}_{i=1}^{n_0} \sim \mathcal{D}_0$
Online Adaptation
new classes
unlabeled data
$t = 1$
$\mathbf{w}_1$
$\{\mathbf{x}_i\}_{i=1}^{n_1} \sim \mathcal{D}_1$
$t = 2$
$\mathbf{w}_2$
$\{\mathbf{x}_i\}_{i=1}^{n_2} \sim \mathcal{D}_2$
$t = 3$
$\mathbf{w}_3$
$\{\mathbf{x}_i\}_{i=1}^{n_3} \sim \mathcal{D}_3$
...
T round

Fig. 1. Illustration of the N-OLS problem protocol. During the offline initialization stage, the learner observes a substantial amount of labeled data from known classes; in the online adaptation stage, the learner receives only a limited amount of unlabeled data, where new classes emerge. Additionally, the data distribution changes over time.

introduce bias to the estimator that is solely trained on known classes. On the other hand, label shifts in known classes data can worsen the identification of new classes. Therefore, it is crucial to adaptively learn the model in the online label shift setting with the emerging new classes.

To handle this problem, we explore the unlabeled data and develop a novel risk estimator that employs risk rewriting and mixture proportion estimation techniques, enabling updates of the model under unknown levels of distribution shift. To adapt to the continuous label shift in data streams, we employ the paradigm of online ensemble [14], which maintains a group of base learners and adaptively combines their outputs to track the changing distribution. Besides, we also introduce the sketching and ensemble pruning mechanisms to improve the computational efficiency of the algorithm, making it more practical for real-world applications. We propose *HAndling New class in Online Label shift* (HANOL) algorithm, which enjoys a theoretical guarantee of dynamic regret, ensuring its effectiveness in adapting to the evolving data distribution and new classes. Empirical experiments are conducted to evaluate the proposed method, including five benchmark datasets, and two real-world applications SHL [15] and fMoW [16]. Our method enhances average accuracy by 10% on SHL and 4% on fMoW datasets, thereby showing its effectiveness for tackling the emerging new classes in online label shift data streams.

Organization: Section II discusses related works. Section III formulates N-OLS problem. Section IV presents our approach. Section V provided theoretical justifications. Section VI reports the experiments. Section VII concludes the paper.

II. Related Work

In the following, we discuss the related topics.

A. Learning Data Streams With Changing Distribution

Supervised Stream with Changing Distributions: The challenge of distribution change is a widely studied topic in the field of streaming data learning [17], [18], [19], [20], [21], [22], [23]. To adapt to the changing distributions, learning approaches can generally be divided into single model-based and ensemble-based approaches. For single model-based approaches, a common practice involves reducing the importance of long-term historical data using techniques such as forgetting mechanisms [24] or windowing mechanisms [25]. Another group of single model-based algorithms enables adaptation to the distribution changes through the detection of such changes. These detectors identify the distribution changing points and subsequently trigger the model to rebuild or update [26]. Recent theoretical advances in online learning show that models can automatically adapt to distribution changes through proper restart mechanisms [27], [28].

Another important category is the ensemble-based model, which has received significant attention in handling distribution change in data streams, which maintains multiple diverse base learners and combines them to get the final prediction. By continuously updating and assigning different weights to base models based on their prediction performances, the ensemble methods can adapt to continuous distribution changes [29], [30]. With a well-designed updating process, ensemble-based algorithms can benefit from solid theoretical guarantees [14], [31]. However, this line of research primarily focuses on the supervised or semi-supervised setting, requiring labeled data to provide timely feedback for the model. As a result, these methods face challenges when handling the unsupervised data stream with distribution changes.

Unsupervised Stream with Online Label Shift: Label shift, as a common type of distribution change, has been extensively studied in the context of "one-step" adaptation [32], [33], where one aims to adapt the model from the source to the target distribution. More recently, the focus has shifted towards scenarios involving streaming data setting where label shifts continuously occur over time [4], [7]. Wu et al. [4] constructed an unbiased risk estimator for the online unlabeled data and employed online gradient descent for model updating. While this preliminary study performs well in scenarios where the label shift in the stream remains unchanged, it faces challenges in non-stationary environments where the class prior can change over time. To tackle this challenge, Bai et al. [7] pioneer the use of the online ensemble framework [14] developed in the modern online learning community to effectively address continuous label shifts with provable guarantees. Nevertheless, these methods do not take into account the challenge of the new class data in the open environments.

B. Classification With New Class Data

New Class Identification: Identification of new class data, or named as open set recognition, is a prominent area of research in computer vision and pattern recognition, focusing on the identification of new classes within a fixed unlabeled dataset [34], [35]. Several methods have been deployed to handle this issue, including nearest neighbor approach [36], adversarial sample generation [37], etc. However, we note that many works in open set recognition implicitly use the feature semantic information to help identify unknown classes. By contrast, we focus on a general setting without such domain knowledge of the semantic information.

Data Stream with New Classes: Learning data streams with new classes requires the learning system to identify the new classes in the unlabeled data stream and adapt the model accordingly [12], [13], [38]. Mu et al. [12] propose an innovative method that leverages an isolation forest [39] to detect emerging new classes and subsequently update the models. Cai et al. [13] propose an ensemble-based nearest neighbor approach to handle scenarios where emerging new classes are not geometrically distant from the known classes. Similarly, Zhang et al. [38] propose a k-nearest neighbor ensemble-based method that explores the neighborhood information to assist in handling new classes. Nevertheless, conventional approaches often overlook the distribution change problem, which can significantly impact both the identification of new class data and the performance of the fixed classifier on known classes.

C. Discussion With Previous Works

Although previous works have studied the data streams with new class and the online label shift problems separately, however, to the best of our knowledge, our work is the first to study the joint problem of the new class with label shift problems especially for the streaming data scenario, and extending the preliminary conference version [40] by considering emerging new classes. The *conjunction of label shift and emerging new classes* is a more challenging problem compared to the two individual problems: the emerging of new class may cause the estimator built on the known classes to be biased, and the label shift problem may cause the new class to be further misclassified. Besides, the learner can only receive unlabeled streams. Consequently, updating the model in the presence of both emerging new class and label shift becomes particularly challenging, and therefore leading to a severe performance drop. In this work, we carefully design a novel risk estimator to handle the emerging new class in online label by exploring the unlabeled data shift via unbiased risk rewriting and mixture proportion estimation techniques, and employ an online ensemble-based paradigm to handle the unknown distribution changes.

III. Problem Formulation

In this section, we formulate the learning problem. We consider a multi-class classification setting. The feature space is denoted by $\mathcal{X} \subseteq \mathbb{R}^d$, where d represents the feature dimension. The label space consists of $K + \mathsf{nc}$ classes in total. Here, within the total label space $\mathcal{Y} = \{1, \dots, K + \mathsf{nc}\}$, classes $[K] \triangleq \{1, \dots, K\}$ represent the known classes in the initial offline labeled data, and $\mathcal{Y}^{\mathsf{nc}} \triangleq \{K + 1, \dots, K + \mathsf{nc}\}$ represents the set of new classes which does not appear in the offline data but emerging in the online unlabeled data streams.

In addition to the presence of the emerging new classes, we consider the occurrence of online label shift. Specifically, throughout the entire time horizon of the unlabeled data stream, conditional distribution remains unchanged, i.e., $\mathcal{D}_t(\mathbf{x} \mid y) = \mathcal{D}_0(\mathbf{x} \mid y)$ for all $\mathbf{x} \in \mathcal{X}, y \in [K]$ and $t \in [T]$; $\mathcal{D}_t(\mathbf{x} \mid y) = \mathcal{D}_{t-1}(\mathbf{x} \mid y)$ for all $\mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}^{\mathsf{nc}}$ and $t \geq 2$. The label distribution can change dynamically, i.e., $\mathcal{D}_t(y = j) \neq \mathcal{D}_{t-1}(y = j)$ for $j \in [K + \mathsf{nc}]$. Additionally, for every $j \in [K]$, we have $\mathcal{D}_0(y = j) > 0$.

In this paper, we formulate the new class in online label shift problem into two phases: the offline supervised initialization and the online unsupervised adaptation, detailed as follows.

- *Offline Supervised Initialization:* In the offline initialization stage, the learner collects a set of labeled data $S_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$ from the offline distribution $\mathcal{D}_0(\mathbf{x}, y)$ defined over the known classes $\mathcal{X} \times [K]$, i.e.,

$$\mathcal{D}_0(\mathbf{x}) = \sum_{j=1}^{K} \left[\boldsymbol{\mu}_{y_0}\right]_j \cdot \mathcal{D}_0^j(\mathbf{x}), \tag{1}$$

  where $[\boldsymbol{\mu}_{y_0}]_j = \mathcal{D}_0(y = j)$ is the label prior for the jth class, $\mathcal{D}_0^j(\mathbf{x}) = \mathcal{D}_0(\mathbf{x} \mid y = j)$ is the marginal distribution of the feature $\mathbf{x}$ over the known class $j \in [K]$. The goal of initialization is to obtain a well-performed initial model $f_0 : \mathcal{X} \mapsto \mathcal{Y}$ that generalizes over the initial distribution $\mathcal{D}_0$, thus acting as a reliable classifier for known classes.
- *Online Unsupervised Adaptation:* After obtaining the initial model f_0, the learner deploys it to a fully unsupervised changing environment. At round $t \in [T]$, the learner can receive a small number of *unlabeled data* $S_t = \{\mathbf{x}_i\}_{i=1}^{n_t}$ drawn from the current distribution $\mathcal{D}_t(\mathbf{x})$. It is important to note that the label distribution in the online adaptation phase comprises not only the known classes $y \in [K]$, but also new classes $y \in \mathcal{Y}^{\mathsf{nc}}$, absent in the offline data, and is changing over time. In our N-OLS, more and more unseen new classes *successively* emerge in the data stream as time evolves, i.e., the new class $y = K + \kappa$ coming after $y = K + \kappa - 1$ for $\kappa \in [\mathsf{nc}]$. The learner aims to sequentially explore the unlabeled data stream to adaptively update the model $\mathbf{w}_t$ and make accurate predictions for each S_t.

IV. Proposed Approach

In this section, we present our approach. To deal with the challenging issue of the conjunction of the continuous label shift and the arrival of the new classes in the N-OLS problem, we develop a risk estimator by risk rewriting and mixture proportion estimation techniques. Then, we proceed to estimate the changing label prior for known classes and the proportion of the new classes in the risk estimator. Finally, we employ the online ensemble structure which aims to deal with the unknown distribution shift in the data stream. The overall protocol of the approach is illustrated in Fig. 2.

A. Risk Estimator for N-OLS Problem

In this part, we propose a novel risk estimator designed for the N-OLS problem, employed to update the model by leveraging both the unlabeled and offline data. In the initialization stage, the model f_0 can be obtained to deal with the known classes. However, during the online adaptation phase, new classes not included in offline data can appear. Besides, the learner can

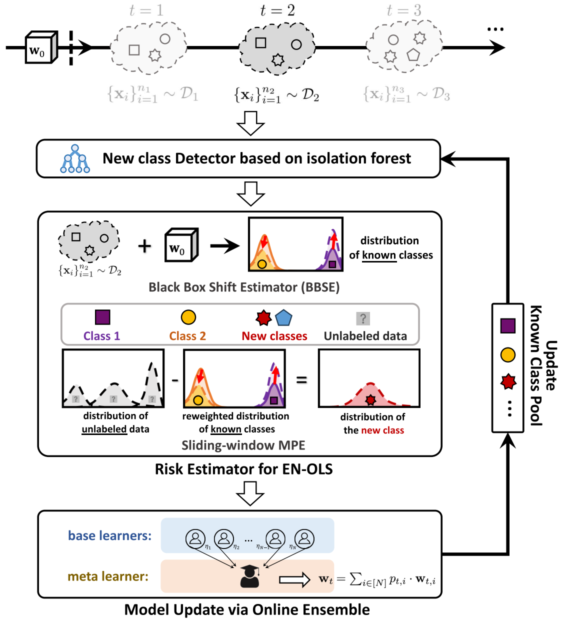

$t=1$
$t=2$
$t=3$
$\mathbf{w}_0$
$\{\mathbf{x}_i\}_{i=1}^{n_1} \sim \mathcal{D}_1$
$\{\mathbf{x}_i\}_{i=1}^{n_2} \sim \mathcal{D}_2$
$\{\mathbf{x}_i\}_{i=1}^{n_3} \sim \mathcal{D}_3$
New class Detector based on isolation forest
distribution of known classes
Black Box Shift Estimator (BBSE)
Class 1
Class 2
New classes
Unlabeled data
distribution of unlabeled data
reweighted distribution of known classes
distribution of the new class
Sliding-window MPE
Risk Estimator for EN-OLS
base learners:
η_1
η_2
η_{N-1}
η_N
meta learner:
$\mathbf{w}_t = \sum_{i\in[N]} p_{t,i} \cdot \mathbf{w}_{t,i}$
Model Update via Online Ensemble
Update Known Class Pool

Fig. 2. Overall protocol of our proposed approach. We first build an isolation forest to detect the emergence of a new class based on the current unlabeled data. Next, we develop a novel risk estimator by exploiting the unlabeled data stream. Subsequently, we employ the online ensemble paradigm to adapt to the continuous label shift. When the number of the encountered new class is large enough, we add this class into the known class pool. This process can be iterated to accommodate the emergence of new classes in the N-OLS data stream, as both the tree-based detector and the classification model can be updated in an online manner.

only obtain a few unsupervised data each round. Consequently, the online risk $R_t(\mathbf{w}) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(f(\mathbf{w},\mathbf{x}),y)]$ is not directly observable as the distribution $\mathcal{D}_t$ is unknown to the learner, where $f(\cdot,\cdot)$ is the prediction function and $\mathbf{w}$ is the model parameter.

We propose a novel risk estimator by exploiting unlabeled data stream via risk rewriting technique. We denote $R_t^k(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x}\,|\,y=k)}[\ell(f(\mathbf{w},\mathbf{x}),k)]$ as the risk of the model over the kth class at round t, where $t \in \{0\} \cup [T]$. Suppose there are K known classes, and the newly emerged class is denoted as $y = K+1$. Then we have $R_t^k(\mathbf{w}) = R_0^k(\mathbf{w})$ for the known classes $k \in [K]$ due to the online label shift assumption $\mathcal{D}_t(\mathbf{x}\,|\,y) = \mathcal{D}_0(\mathbf{x}\,|\,y)$. However, since new classes can emerge in the online unlabeled data stream, label distribution of the new class is unavailable, making the new class risk $R_0^{K+1}(\mathbf{w})$ unknown. To tackle this issue, we propose a novel risk estimator for the expected online risk R_t. We first notice that the marginal distribution $\mathcal{D}_t(\mathbf{x})$ can be decomposed as

$$\begin{aligned}(1-\theta_t)\mathcal{D}_t^{K+1}(\mathbf{x}) &= \mathcal{D}_t(\mathbf{x}) - \theta_t\mathcal{D}_t^{\mathsf{kc}}(\mathbf{x})\\ &= \mathcal{D}_t(\mathbf{x}) - \theta_t\left(\sum_{j=1}^{K}[\boldsymbol{\mu}_{y_t}]_j\mathcal{D}_t(\mathbf{x}\,|\,j)\right)\\ &= \mathcal{D}_t(\mathbf{x}) - \theta_t\left(\sum_{j=1}^{K}[\boldsymbol{\mu}_{y_t}]_j\mathcal{D}_0(\mathbf{x}\,|\,j)\right),\end{aligned} \tag{2}$$

where $\mathcal{D}_t^{K+1}$ is the distribution of the new class data in $\mathcal{D}_t$, $\mathcal{D}_t^{\mathsf{kc}}$ is the distribution of known classes in $\mathcal{D}_t$, $\boldsymbol{\mu}_{y_t} \in \Delta_K$ is the label distribution vector of known classes, and $(1-\theta_t) \in [0,1]$ is the proportion of the new class at round t. The first two equations in (2) are derived using the law of total probability, while the final equation is obtained with the label shift assumption. By (2), we focus on the new class risk where $y = K+1$ and rewrite the new class risk $R_t^{K+1}(\mathbf{w})$ as

$$\begin{aligned}(1-\theta_t)R_t^{K+1}(\mathbf{w}) &\triangleq (1-\theta_t)\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t^{K+1}(\mathbf{x})}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right]\\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x})}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right]\\ &\quad - \theta_t\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t^{\mathsf{kc}}(\mathbf{x})}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right]\\ &= \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x})}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right]\\ &\quad - \theta_t\sum_{j=1}^{K}\left[\boldsymbol{\mu}_{y_t}\right]_j\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_0^j(\mathbf{x})}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right].\end{aligned}$$

The expected risk over distribution $\mathcal{D}_t(\mathbf{x})$ can be approximated by the empirical risk over the unlabeled data S_t, given by $1/n_t\sum_{\mathbf{x}\in S_t}\ell(f(\mathbf{w},\mathbf{x}),K+1)$, while the risk over $\mathcal{D}_0^j(\mathbf{x}) \triangleq \mathcal{D}_0(\mathbf{x}\,|\,y=j)$ can be approximated by empirical risk over offline data S_0. Hence, we can build an estimator $\widehat{R}_t(\mathbf{w})$ for the expected risk $R_t(\mathbf{w})$ as follows:

$$\begin{aligned}\widehat{R}_t(\mathbf{w}) &= \frac{1}{n_t}\sum_{(\mathbf{x}_i,y_i)\in S_t}\ell(f(\mathbf{w},\mathbf{x}_i),y_i)\\ &= \widehat{\theta}_t\widehat{R}_t^{\mathsf{kc}}(\mathbf{w}) + (1-\widehat{\theta}_t)\widehat{R}_t^{K+1}(\mathbf{w})\\ &= \widehat{\theta}_t\sum_{j=1}^{K}\left[\widehat{\boldsymbol{\mu}}_{y_t}\right]_j R_0^j(\mathbf{w}) + \sum_{\mathbf{x}\in S_t}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right]\\ &\quad - \widehat{\theta}_t\sum_{j=1}^{K}\left[\widehat{\boldsymbol{\mu}}_{y_t}\right]_j\sum_{\mathbf{x}\in S_0^j}\left[\ell\left(f\left(\mathbf{w},\mathbf{x}\right),K+1\right)\right].\end{aligned} \tag{3}$$

Overall, we build a risk estimator $\widehat{R}_t(\mathbf{w})$ by leveraging online unlabeled data and offline labeled data. The remaining question is how to estimate the parameters $\widehat{\theta}_t$ and $\widehat{\boldsymbol{\mu}}_{y_t}$. In the following, we use *black box shift estimator* (BBSE) [33] to estimate the label distribution $\boldsymbol{\mu}_{y_t}$, and employ *mixture proportion estimation* (MPE) methods [41], [42] to estimate θ_t given that we can empirically observe $\mathcal{D}_0(\mathbf{x}\,|\,y=j)$ and $\mathcal{D}_t(\mathbf{x})$.

B. Label Distribution Estimation With Unlabeled Data

In this part, we introduce the details of how to estimate the changing label distribution $\boldsymbol{\mu}_{y_t}$ for known classes, and the proportion of the new class θ_t. In addition to the simple case of handling a single new class $y = K+1$, we further illustrate how we handle *emerging new classes* by employing a tree-based new class detector.

Estimate Proportion for Known Classes: We use BBSE to estimate the class prior of the known classes $\boldsymbol{\mu}_{y_t}$ via solving

$$\widehat{\boldsymbol{\mu}}_{y_t} = C_0^{-1}\cdot\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}, \tag{4}$$

where $\widehat{\boldsymbol{\mu}}_{\widehat{y}_t} \in \Delta_K$ with $[\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}]_j = 1/n_t \cdot \sum_{\mathbf{x} \in S_t} [f_0(\mathbf{x})]_j$ is the estimated class prior of the prediction $f_0(\mathbf{x})$, and $C_0 \in \mathbb{R}^{K \times K}$ is the classifier's confusion matrix with $[C_0]_{i,j} \triangleq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_0(\mathbf{x} \mid y=j)}[\mathbb{1}(f_0(\mathbf{x}) = i)]$ being the classification rate that the initial model f_0 predicts samples from class i as class j. Benefit from the benign properties of BBSE, we can guarantee that the estimation $\widehat{\boldsymbol{\mu}}_{y_t}$ satisfies $\mathbb{E}[\widehat{\boldsymbol{\mu}}_{y_t}] = \boldsymbol{\mu}_{y_t}$, where the ground-truth label distribution is $\boldsymbol{\mu}_{y_t} \triangleq C_0^{-1} \boldsymbol{\mu}_{\widehat{y}_t} = \mathcal{D}_t^{\mathsf{kc}}(y)$. We assume the offline data to be sufficient and of high quality, ensuring that the estimated known class distribution remains accurate and stable, even when encountering new classes.

Estimate Proportion for New Class: Notice that the construction of the risk estimator $\widehat{R}_t$ requires estimating the proportion θ_t, which corresponds to the problem of MPE, where one aims to estimate the proportion of a certain class within the overall distribution based on empirical observations.

If the sample size is sufficiently large in each round, the proportion of the new class can be estimated using the existing MPE technique. However, in online scenarios, the amount of data obtained in each round is minimal, and direct estimation can lead to high variance. To address this issue, we propose a sliding window-based MPE algorithm. Specifically, we maintain a window queue of length L. At time t, following the first-in-first-out principle, the current round sample S_t is added to the queue, and a certain number of samples are removed from the queue's front. At each time step, we utilize the samples in the sliding window to estimate the proportion of known classes. Specifically, inspired by the recently proposed *Best Bin Estimation* (BBE) technique [43], we utilize resampled offline labeled data and the unlabeled data in the sliding window to estimate the proportion of the new class. We first train a well-performed classifier $h(\cdot) : \mathcal{X} \mapsto [0, 1]$, where 0 means the data is sampled from the online unlabeled data and 1 means the data is sampled from labeled data, then we get

$$q_p(z) = \frac{\sum_{\mathbf{x}_i \in S_0} \mathbb{I}[h(\mathbf{x}_i) \geqslant z]}{|S_0|}, q_u(z) = \frac{\sum_{\mathbf{x}_i \in S_{\text{win}}} \mathbb{I}[h(\mathbf{x}_i) \geqslant z]}{|S_{\text{win}}|},$$

where S_0 is the offline dataset, and S_{win} is the online unlabeled data in the sliding window. By solving the equation

$$\widehat{c} = \arg\min_{c \in [0,1]} \times \left\{ \frac{q_u(c)}{q_p(c)} + \frac{1+\gamma}{q_p(c)} \left(\sqrt{\frac{\log(4/\delta)}{2S_{\text{win}}}} + \sqrt{\frac{\log(4/\delta)}{2S_0}} \right) \right\},$$

where $0 < \delta, \gamma < 1$ are the hyperparameters, then, we can estimate the proportion of the new class by

$$\widehat{\theta}_t = q_u(\widehat{c}) / q_p(\widehat{c}).$$

The estimation of the new class proportion by the sliding-window MPE method enjoys a convergence rate of $\mathcal{O}(\min(|S_0|, |S_{\text{win}}|)^{-1/2})$ [43], therefore can obtain the proportion of new class with a small variance.

Detect the Emerging New Classes: Note that our method is applicable to handle the emerging new classes [3], where more and more unseen new classes successively arise *one after the other* as time evolves. A key component is the *new class detector*, for which we utilize a tree-based detection approach [12], which detects the new class based on the isolation forest [39]. Specifically, this detector recognizes a new (previously unseen) class by measuring the isolation depth of the samples within the forest. Typically, the distribution of the newly emerged class samples differs significantly from that of known classes, resulting in a higher isolation depth for new class samples, thereby enabling effective detection. We maintain a buffer to store the sketched new class samples in the data stream and detect whether another new class has emerged. Then, we update the model parameter $\mathbf{w}_t$ using previously seen classes and the new class by our proposed risk estimator $\widehat{R}_t(\mathbf{w})$. As a result, the previously detected new class transits to a known class, and we add it into the known class pool. This process can be iterated to accommodate emerging new classes, regardless of the number of new classes, and the model $\mathbf{w}_t$ can be updated in an online manner. Note that this detection method aims to identify new classes that emerge sequentially, and the more challenging problem of detecting multiple new classes simultaneously is left for future work.

C. Adaptation via Online Ensemble

Based on the risk estimator $\widehat{R}_t(\mathbf{w})$ constructed in Section IV-A and the parameter estimating approach in Section IV-B, we then design an online algorithm to adapt the model $\mathbf{w}_t$ to the changing distribution $\mathcal{D}_t$. A natural choice is to minimize the risk estimator $\widehat{R}_t(\mathbf{w})$ from scratch, which means $\mathbf{w}_t \in \arg\min_{\mathbf{w}} \widehat{R}_t(\mathbf{w})$. Whereas, $\widehat{R}_t(\mathbf{w})$ can suffer from high variance due to the very small online sample size n_t, which may lead to poor generalization performance. To this end, we turn to reuse historical information via *online gradient descent* (OGD). However, OGD with a fixed step size may not be able to adapt to the changing distribution. To handle this issue, we propose an adaptive online ensemble algorithm with a two-layer structure, which can adaptively track the suitable step size, as shown in Algorithm 1.

In order to adapt to the changing distributions, we employ the paradigm of online ensemble learning [14]. More specifically, as demonstrated in Fig. 2, we maintain a set of *base learners* that are updated with different step sizes, corresponding to varying intensity of label distribution variations. Simultaneously, we maintain an *meta learner* that integrates outputs of these base learners, enabling adaptive tracking of the optimal base learner and therefore handling the challenge of online distribution changes.

- *Construct base learners with multiple step sizes:* At round t, with risk estimator $\widehat{R}_t(\mathbf{w})$ in (3), we can obtain the estimated gradient $\nabla \widehat{R}_t(\mathbf{w})$ and update the model $\mathbf{w}_t$ by gradient descent, given by the following update schedule

  $$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}} \left[\mathbf{w}_t - \eta \nabla \widehat{R}_t(\mathbf{w}_t) \right],$$

  where $\Pi_{\mathcal{W}}[\cdot]$ denotes the projection onto the parameter domain $\mathcal{W}$ and η is the step size. This OGD algorithm benefits from the risk estimator $\widehat{R}_t(\mathbf{w})$ that can evaluate the model properly for all classes, including the emerging new classes in online streams, which enables continuous updates of the model in the correct direction.

Algorithm 1: HANOL: HAndling New Class in OLS.

```
Require: step size pool $\mathcal{H}$; learning rate $\varepsilon$; step size $\eta_i \in \mathcal{H}$
initialization: get $\mathbf{w}_1^i \in \mathcal{W}$ by offline supervised
    initialization; $\forall i \in [N], p_1^i = 1/N$
for $t = 2$ to $T$ do
    for $i = 1$ to $N$ do
        construct risk estimator $\widehat{R}_t(\mathbf{w}_t^i)$ as (3)
        update the $i$th base model $\mathbf{w}_t^i$ by (5)
        update the weight $p_t^i$ according to (6)
    end for
    output final model $\mathbf{w}_t = \sum_{i=1}^N p_t^i \cdot \mathbf{w}_t^i$
end for
```

However, the OGD algorithm with a single step size may have difficulty in adapting to the changing distribution $\mathcal{D}_t$ – ideally, the step size η should be small when the distribution changes slowly and large when it changes rapidly. The key challenge is to adaptively track the appropriate step size η_t without prior knowledge of the distribution shifts. Therefore, inspired by ensemble learning, we propose an online ensemble algorithm to adaptively track the suitable step size η_t to the distribution $\mathcal{D}_t$. Specifically, we maintain a set of base learners with different step sizes $\mathcal{H} = \{\eta_i\}_{i=1}^N$. At round t, we update the ith base learner $\mathbf{w}_t^i$ by

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}}\left[\mathbf{w}_t^i - \eta_i \nabla \widehat{R}_t\left(\mathbf{w}_t^i\right)\right]. \tag{5}$$

Thus, we can obtain a set of base models $\{\mathbf{w}_t^i\}_{i=1}^N$, where different base models excel in handling the online label shift of different intensities.

- *Combine the outputs by meta learner:* We maintain a meta learner that combines the outputs of multiple base learners through weighted averaging to obtain the final model, given by $\mathbf{w}_t = \sum_{i=1}^N p_t^i \cdot \mathbf{w}_t^i$. The weights $p_t^i \in [0, 1]$ denotes the extent of utilization for the ith base learner, and statistically satisfies $\sum_{i=1}^N p_t^i = 1$. Since the environment changes dynamically as time evolves, the weights p_t^i should be adaptively updated according to the current performance of each base learner, that is,

$$p_t^i \propto \exp\left(-\varepsilon \sum_{s=1}^{t-1} \widehat{R}_s(\mathbf{w}_s^i)\right), \tag{6}$$

where $\varepsilon > 0$ is a hyperparameter that controls the sensitivity of the meta learner to the performance of base learners. Intuitively, the meta learner assigns higher weights to base learners that exhibit better cumulative performance, i.e., smaller cumulative risks, thereby enabling adaptive tracking of the optimal base learner.

In Section V, we will theoretically demonstrate that the proposed online ensemble algorithm can track the optimal base learner adaptively by only maintaining about $\log T$ learners.

D. Efficiency Consideration

In this section, we discuss how to improve the computational and storage efficiency of the proposed algorithm. Specifically, we introduce a sketching technique named balanced kernel herding to store the offline dataset more efficiently, and propose an ensemble pruning mechanism for reducing the number of base learners in our HANOL to improve the computational efficiency.

Sketching the Offline Dataset: As we mentioned in Sections IV-A and IV-B, we have built a novel risk estimator for the N-OLS problem, and estimate the hyperparameter using BBSE and our sliding window-based MPE algorithm. However, we note that these methods need to store and re-calculate the entire offline data S_0 at each time step, which is computationally expensive and memory-consuming. To address this issue, we use a subset of samples to "sketch" the offline data, which efficiently approximates the distribution of the offline data S_0. Specifically, we propose the *balanced kernel herding* mechanism to extract sketches of the offline dataset, which is inspired by Chen et al. [44] and Wu et al. [45]. Our balanced kernel herding is a deterministic, iterative algorithm that samples informative points in the dataset. For each known class $k \in [K]$, we run the following two steps to sketch the offline data S_0^k in the kth class:

$$\begin{aligned}\mathbf{s}_i^k &= \operatorname*{arg\,max}_{\mathbf{s}\in\mathcal{X}} \left\langle \psi_i^k, \phi(\mathbf{s})\right\rangle, \\ \psi_{i+1}^k &= \psi_i^k + \mu\left(S_0^k\right) - \phi\left(\mathbf{s}_i^k\right);\end{aligned} \tag{7}$$

where μ is the kernel mean embedding function, i.e., $\mu(S_0^k) := \frac{1}{|S_0^k|}\sum_{i=1}^{|S_0^k|}\phi(\mathbf{x}_i)$ and ϕ is a feature mapping associated with the positive definite symmetric kernel. Here we choose the Gaussian kernel, i.e., $\langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle = \mathrm{e}^{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2}}$, where σ is a user-specific hyperparameter to control the kernel width. The initial ψ_1 is set as $\mu(S_0)$. To sketch the offline dataset S_0, we iterate through (7) for m steps to get a sketched set $\{\mathbf{s}_1^k, \ldots, \mathbf{s}_m^k\}$ for each class $k \in [K]$. The risk estimated by our sketched samples enjoys a good convergence rate compared with the original offline dataset, as detailed in Section V. Therefore, it can effectively preserve the distribution information of the offline data S_0 using only mK samples. When detecting a newly emerged class, we can also use the same sketching technique to sketch new class samples.

Ensemble Pruning: Our proposed HANOL method employs the paradigm of online ensemble [14]. However, this typically needs to update a total of $\mathcal{O}(\log T)$ base learners per round, which may be costly as time grows and some efforts have been made to improve the projection efficiency [46]. To this end, inspired by previous ensemble pruning methods [47], [48], we employ the *ordering-based* pruning mechanism to reduce the number of base learners. Specifically, at each round, we only maintain the most accurate N base learners, i.e., pruning the base learners according to their order of the cumulative historical accuracy. We then combine these selected base learners and use a meta learner to get the final output, thereby reducing the number of base learners and improving the computational efficiency of our algorithm.

V. Theoretical Results

In this section, we analyze the theoretical properties of our algorithm. We introduce and employ *dynamic regret* [49] as the theoretical performance measure. Following that, we first present the theoretical analysis with detailed discussions, then we provide the corresponding proofs.

A. Theoretical Analysis

We consider the convex flexible domain and loss functions. Our goal is to obtain a sequence of online model parameters $\{\mathbf{w}_t\}_{t=1}^T$ that can minimize the cumulative expected risk over the whole time horizon: $\sum_{t=1}^T R_t(\mathbf{w}_t)$. The excepted risk $R_t(\mathbf{w})$ at each round is defined as $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(f(\mathbf{w},\mathbf{x}),y)]$, where $\ell : \mathbb{R}^{K+\mathsf{nc}} \times \mathcal{Y} \mapsto \mathbb{R}$ is any convex loss function and $f(\mathbf{w},\mathbf{x})$ is the prediction of the model $\mathbf{w}\in\mathcal{W}$ on the feature $\mathbf{x}$. We adopt the dynamic regret $\text{Reg}_T^{\mathbf{d}}$ as the performance measure [7], [50]. It is defined as the difference between the cumulative expected risk of the predictive model sequence $\{\mathbf{w}_t\}_{t=1}^T$ and the model sequence $\{\mathbf{w}_t^\star\}_{t=1}^T$:

$$\text{Reg}_T^{\mathbf{d}} \triangleq \sum_{t=1}^T R_t(\mathbf{w}_t) - \sum_{t=1}^T R_t(\mathbf{w}_t^\star),$$

where the model parameter $\mathbf{w}_t^\star \in \arg\min_{\mathbf{w}\in\mathcal{W}} R_t(\mathbf{w})$ we defined in this paper is the best model at each round t. A small dynamic regret indicates that the proposed algorithm can adapt to a changing environment and achieve a performance competitive with the best model sequence.

We denote the upper bound of gradient norm by $G \triangleq \sup_{\mathcal{X},\mathcal{Y},\mathcal{W}} \|\nabla\ell(f(\mathbf{w},\mathbf{x}),y)\|_2$ and the diameter of the convex parameter space $\mathcal{W}$ by $\Gamma \triangleq \sup_{\mathbf{w}_1,\mathbf{w}_2\in\mathcal{W}} \|\mathbf{w}_1-\mathbf{w}_2\|_2$. We use $B \triangleq \sup_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y},\mathbf{w}\in\mathcal{W}} |\ell(f(\mathbf{w},\mathbf{x}),y)|$ as the upper bound of loss function value, and σ as the minimum singular value of the confusion matrix C_0. Under the assumption that the confusion matrix C_0 is invertible, i.e., $\sigma>0$, our algorithm enjoys the following dynamic regret guarantee.

Theorem 1 (Dynamic Regret): Suppose the confusion matrix C_0 is invertible. Set the step size pool as $\mathcal{H}=\{\eta_i=\frac{\sigma\Gamma}{2G\sqrt{(K+1)T}}\cdot 2^{i-1} \mid i\in[N]\}$, where $N=1+\lceil\frac{1}{2}\log_2(1+2T)\rceil$ is the number of the base learners. Our proposed HANOL ensures that

$$\mathbb{E}\left[\text{Reg}_T^{\mathbf{d}}\right] \le \mathcal{O}\left(\max\left\{V_T^{1/3}T^{2/3}, \sqrt{T}\right\}\right),$$

or simplified as $\mathcal{O}(V_T^{1/3}T^{2/3})$ for non-degenerated cases of $V_T \ge \Theta(T^{-\frac{1}{2}})$, where $V_T=\sum_{t=2}^T \|\mathcal{D}_t(y)-\mathcal{D}_{t-1}(y)\|_1$ measures the intensity of label distributions variation.

Proof Sketch: To prove Theorem 1, we require the unbiasedness of estimating the class priors in the risk estimator given by (3) for the unlabeled data stream. This unbiasedness enables us to transform the N-OLS problem back into the non-stationary online learning problem. We propose efficient approximations for the unbiased estimator of class priors. To estimate the prior of new class data, we employ the MPE technique based on offline data and a sliding window of online data. To estimate the label distribution for known classes, we handle it by a well-performed classifier f_0 trained on offline data and the confusion matrix C_0, where we assume the offline data to be adequately collected, such that $y\in[K]$, $\mathcal{D}_0(y)>0$ and therefore the obtained confusion matrix C_0 is invertible. The detailed proof of Theorem 1 is provided in Section V-B.

Remark 1: The non-stationarity inherent in the N-OLS problem arises from the label space $\mathcal{Y}$, which encompasses the continuous change in label distribution of known classes and the arrival of the new class. As shown in Theorem 1, the expected dynamic regret is impacted by the magnitude of label distribution variations. It is worth mentioning that the presence of new classes can be regarded as a specific type of label distribution variation, where the prior probabilities of the new class transition from 0 to non-zero values. Consequently, the intensity of class-prior variation V_T in Theorem 1 characterizes both the label distribution variations of known classes and the emergence of new classes, indicating that our proposed algorithm can adapt to the online changing environment without the prior knowledge of the distribution shift intensity.

Theorem 2 (Efficiency): By employing the proposed sketching and ensemble pruning mechanisms, the overall computational complexity of our algorithm is reduced from $\mathcal{O}(|S_0|\cdot\log T)$ to $\mathcal{O}(m(K+\mathsf{nc})N)$ per round, where $m(K+\mathsf{nc}) \ll |S_0|$ is the sample size of the sketched dataset, and N is the number of the selected base learners. Meanwhile, the proposed efficiency mechanism also enjoys benign theoretical guarantees, introducing only an extra error of $\mathcal{O}(1/m)$ to estimate the risk each round under certain assumptions.

B. Proof of Theorem 1

Proof of Theorem 1: To prove Theorem 1, we first show that the constructed risk estimator enjoys the benign property of unbiasedness under certain conditions. Then, we analyze the dynamic regret of our proposed algorithm by decomposing it into two components: *base regret* and *meta regret*. We provide the detailed proof in the following.

Unbiasedness: We first introduce the following lemma to show that our designed risk estimator enjoys the benign property of unbiasedness, providing reliable guidance for model updates.

Lemma 1 (Unbiased Risk Estimator): The proposed risk estimator $\widehat{R}_t(\mathbf{w})$ in (3) is unbiased to $R_t(\mathbf{w})=\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(f(\mathbf{w},\mathbf{x}),y)]$, i.e., $\mathbb{E}_{S_t\sim\mathcal{D}_t}[\widehat{R}_t(\mathbf{w})]=R_t(\mathbf{w})$, for any $\mathbf{w}\in\mathcal{W}$ independent of the dataset S_t, provided C_{f_0} is invertible and the offline dataset S_0 has sufficient samples such that $\widehat{C}_{f_0}=C_{f_0}$ and $\widehat{R}_0^k(\mathbf{w})=R_0^k(\mathbf{w}), \forall k\in[K+\mathsf{nc}]$.

Proof of Lemma 1: First, we show that the BBSE's estimation $\widetilde{\boldsymbol{\mu}}_t=C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}$ is unbiased towards the ground truth label prior $\widetilde{\boldsymbol{\mu}}_t$ if the initial data is sufficient such that we can obtain C_0. We rewrite the BBSE's estimation as $\widetilde{\boldsymbol{\mu}}_t=C_0^{-1}\widetilde{\boldsymbol{\mu}}_{\widehat{y}_t}=C_0^{-1}\frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t} h_0(\mathbf{x})$. Taking the expectations of both sides,

$$\begin{aligned}\mathbb{E}_{S_t\sim\mathcal{D}_t}\left[\widetilde{\boldsymbol{\mu}}_t\right] &= \mathbb{E}_{S_t\sim\mathcal{D}_t}\left[C_0^{-1}\left(\frac{1}{|S_t|}\sum_{\mathbf{x}\in S_t} h_0(\mathbf{x})\right)\right]\\ &= \mathbb{E}_{S_t\sim\mathcal{D}_t}\left[C_0^{-1}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t(\mathbf{x})}\left[h_0(\mathbf{x})\right]\right]=C_0^{-1}\boldsymbol{\mu}_{\widehat{y}_t}=\boldsymbol{\mu}_t,\end{aligned}$$

which shows that the BBSE's estimation is unbiased towards the ground truth label prior $\boldsymbol{\mu}_t$. Besides, for our Sliding-window MPE, if we have sufficient online unlabeled data and high-quality offline labeled data, the estimated new class proportion

$\widehat{\theta}_t$ converges to the true value θ_t [43]. Therefore, we finish the proof of Lemma 1. □

Then, by leveraging the unbiased properties of the risk estimator, we analyze the dynamic regret of our method. Specifically, we convert the overall dynamic regret into two components: meta regret and base regret.

$$\underbrace{\sum_{t=1}^{T}\Big(R_t(\mathbf{w}_t)-R_t(\mathbf{w}_t^i)\Big)}_{\text{meta regret}}+\underbrace{\sum_{t=1}^{T}\Big(R_t(\mathbf{w}_t^i)-R_t(\mathbf{w}_t^\star)\Big)}_{\text{base regret}}.$$

Base Regret: The base regret measures the gap between the base model and the optimal model sequence. To further examine the base regret, following Bai et al. [7], we introduce a piecewise stationary reference sequence that changes every Δ iterations, and decompose base regret into two parts:

$$\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}R_t(\mathbf{w}_t)\right]-\sum_{t=1}^{T}R_t(\mathbf{w}_t^\star)\\
&=\underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}R_t(\mathbf{w}_t)\right]-\sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m}R_t(\mathbf{w}_{\mathcal{I}_m}^\star)}_{\text{term (a)}}\\
&\quad+\underbrace{\sum_{m=1}^{M}\sum_{t\in\mathcal{I}_m}R_t(\mathbf{w}_{\mathcal{I}_m}^\star)-\sum_{t=1}^{T}R_t(\mathbf{w}_t^\star)}_{\text{term (b)}},
\end{aligned}$$

where $\mathbb{E}_{1:T}[\cdot]$ denotes the expectation taken over the random draw of dataset $\{S_t\}_{t=1}^T$, and $M=\lceil\frac{T}{\Delta}\rceil\leq T/\Delta+1$ is the number of the intervals. The first part means the gap between the base model sequence and the reference sequence, and the second part means the gap between the reference sequence and the optimal model sequence. Then, we turn to analyze the term (a) and the term (b), respectively.

We first show that the expected risk $R_t(\cdot)$ can be related to the empirical risk estimator $\widehat{R}_t(\cdot)$ due to its unbiasedness property as stated in Lemma 1.

$$\begin{aligned}
\mathtt{term}\ (\mathtt{a})&\leq\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\langle\nabla R_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}_{\mathcal{I}}^\star\rangle\right]\\
&=\underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\langle\nabla R_t(\mathbf{w}_t)-\nabla\widehat{R}_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}_{\mathcal{I}}^\star\rangle\right]}_{\mathtt{term}\ (\mathtt{a}_1)}\\
&\quad+\underbrace{\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\langle\nabla\widehat{R}_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}_{\mathcal{I}}^\star\rangle\right]}_{\mathtt{term}\ (\mathtt{a}_2)},
\end{aligned}$$

where the first inequality is due to the convexity of the risk function $R_t(\cdot)$. Further, for $\mathtt{term}\ (\mathtt{a}_1)$, we have

$$\begin{aligned}
\mathtt{term}\ (\mathtt{a}_1)&=\mathbb{E}_{1:T}\left[\langle\nabla R_t(\mathbf{w}_t)-\nabla\widehat{R}_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}_{\mathcal{I}}^\star\rangle\right]\\
&=\mathbb{E}_{1:t\text{-}1}\Big[\langle\nabla R_t(\mathbf{w}_t)-\mathbb{E}_t[\nabla\widehat{R}_t(\mathbf{w}_t)|1:t-1],\mathbf{w}_t-\mathbf{w}_{\mathcal{I}}^\star\rangle\Big]=0,
\end{aligned}$$

where the last equality is due to the unbiasedness of the risk estimator $\widehat{R}_t$ as stated in Lemma 1, such that $\nabla R_t(\mathbf{w}_t)=\mathbb{E}_t[\nabla\widehat{R}_t(\mathbf{w}_t)\mid 1:t-1]$. Thus, it is sufficient to analyze $\mathtt{term}\ (\mathtt{a}_2)$ to provide an upper bound for $\mathtt{term}\ (\mathtt{a})$. To bound $\mathtt{term}\ (\mathtt{a}_2)$, we give the following useful lemma.

Lemma 2 (Lemma 6 in Bai et al. [7]): For an unbiased risk estimator $\widehat{R}_t(\mathbf{w})$, under same assumptions of Theorem 1, the base regret of one base learner updated by (5) satisfies

$$\sum_{t=1}^{T}\left\langle\nabla\widehat{R}_t(\mathbf{w}_t),\mathbf{w}_t-\mathbf{u}_t\right\rangle\leq\frac{2\eta KG^2T}{\sigma^2}+\frac{2\Gamma P_T+\Gamma^2}{2\eta}$$

for any comparator sequence $\{\mathbf{u}_t\}_{t=1}^T$ with $\mathbf{u}_t\in\mathcal{W}$, where $P_T=\sum_{t=2}^T\|\mathbf{u}_t-\mathbf{u}_{t-1}\|_2$ measures the variation of the comparator sequence.

Since the comparator sequence in $\mathtt{term}\ (\mathtt{a})$ only changes $M-1$ times, its variation is bounded by $P_T\leq\Gamma(M-1)\leq(\Gamma T)/\Delta$. By Lemma 2 and taking the expectation, we have

$$\mathtt{term}\ (\mathtt{a}_2)\leq\frac{2\eta KG^2T}{\sigma^2}+\frac{2\Gamma^2T/\Delta+\Gamma^2}{2\eta}.$$

Combining upper bounds of $\mathtt{term}\ (\mathtt{a}_1)$ and $\mathtt{term}\ (\mathtt{a}_2)$ yields

$$\begin{aligned}
\mathtt{term}\ (\mathtt{a})&\leq\mathtt{term}\ (\mathtt{a}_1)+\mathtt{term}\ (\mathtt{a}_2)\\
&\leq\frac{2\eta KG^2T}{\sigma^2}+\frac{2\Gamma^2T/\Delta+\Gamma^2}{2\eta}.
\end{aligned}$$

Meta Regret: For the meta regret, we have

$$\begin{aligned}
&\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}R_t\left(\mathbf{w}_t\right)-\sum_{t=1}^{T}R_t\left(\mathbf{w}_t^i\right)\right]\\
&\leq\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}R_t\left(\mathbf{w}_{t,j}\right)-\sum_{t=1}^{T}R_t\left(\mathbf{w}_t^i\right)\right]\\
&=\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}\widehat{R}_t\left(\mathbf{w}_{t,j}\right)-\sum_{t=1}^{T}\widehat{R}_t\left(\mathbf{w}_t^i\right)\right.\\
&\quad+\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}\left(R_t\left(\mathbf{w}_{t,j}\right)-\widehat{R}_t\left(\mathbf{w}_{t,j}\right)\right)\right.\\
&\quad\left.+\sum_{t=1}^{T}\left(R_t\left(\mathbf{w}_t^i\right)-\widehat{R}_t\left(\mathbf{w}_t^i\right)\right)\right]\\
&=\mathbb{E}_{1:T}\left[\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}\widehat{R}_t\left(\mathbf{w}_{t,j}\right)-\sum_{t=1}^{T}\widehat{R}_t\left(\mathbf{w}_t^i\right)\right],
\end{aligned}$$

where the first inequality is due to Jensen's inequality and the last equality is due to unbiasedness of our estimator $\widehat{R}_t(\mathbf{w})$. Then, we can upper bound the meta regret as follows.

Lemma 3 (Meta Regret): By setting the learning rate $\varepsilon=\frac{\sigma}{B}\sqrt{\frac{\ln N+2}{(K+\mathsf{nc})T}}$, the meta regret of HANOL satisfies

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{j=1}^{N}p_{t,j}\widehat{R}_t\left(\mathbf{w}_{t,j}\right)-\sum_{t=1}^{T}\widehat{R}_t\left(\mathbf{w}_t^i\right)\\
&\leq\frac{2B}{\sigma}\sqrt{(\ln N+2)(K+\mathsf{nc})T}
\end{aligned}$$

for any $i \in [N]$, where B is the upper bound of the loss function defined as $B \triangleq \sup_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y},\mathbf{w}\in\mathcal{W}} |\ell(f(\mathbf{w},\mathbf{x}),y)|$.

Proof of Lemma 3: Since our HANOL takes the Hedge algorithm as the meta algorithm, we can exploit the standard analysis of Hedge to upper bound the meta regret. We first give the following regret guarantee of the vanilla Hedge algorithm,

Lemma 4 (Theorem 19 of Syrgkanis et al. [51]): Let $\ell_t \in \mathbb{R}^N$ be the loss vector and take $\ell_t^i \in \mathbb{R}$ as its ith entry, the Hedge algorithm updating with $p_t^i \propto \exp(-\varepsilon(\sum_{s=1}^{t-1} \ell_s^i))$ satisfies

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\ell_t^j - \sum_{t=1}^{T}\ell_t^i \le \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}\|\boldsymbol{\ell}_t\|_\infty^2$$

for any $i \in [N]$, where $\varepsilon > 0$ is the step size.

By Lemma 4, we bound the meta regret by

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t\left(\mathbf{w}_{t,j}\right) - \sum_{t=1}^{T}\widehat{R}_t\left(\mathbf{w}_t^i\right)$$
$$\le \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}\left|\max_{i\in[N]}\left\{\widehat{R}\left(\mathbf{w}_t^i\right)\right\}\right|^2, \qquad (8)$$

where the last term can be further bounded by

$$\left|\widehat{R}_t\left(\mathbf{w}_t^i\right)\right| = \left|\sum_{k=1}^{K+\mathsf{nc}}\left[C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right]_k \cdot R_0^k\left(\mathbf{w}_t^i\right)\right|$$
$$\le B\left\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right\|_1 \le B\sqrt{K+\mathsf{nc}}\left\|C_{f_0}^{-1}\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right\|_2$$
$$\le B\sqrt{K+\mathsf{nc}}\left\|C_{f_0}^{-1}\right\|_2\left\|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\right\|_2 \le \frac{B\sqrt{K+\mathsf{nc}}}{\sigma}. \qquad (9)$$

In above equations, the third inequality is due to the Cauchy-Schwarz inequality. The last inequality comes from $\|C_{f_0}^{-1}\|_2 \le \sigma^{-1}$ and $\|\widehat{\boldsymbol{\mu}}_{\widehat{y}_t}\|_2 \le 1$. Plugging (9) into (8), we have

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{t,j}\widehat{R}_t\left(\mathbf{w}_{t,j}\right) - \sum_{t=1}^{T}\widehat{R}_t\left(\mathbf{w}_t^i\right)$$
$$\le \frac{\ln N + 2}{\varepsilon} + \varepsilon\sum_{t=1}^{T}\frac{B^2(K+\mathsf{nc})T}{\sigma^2}.$$

Setting $\varepsilon = \frac{\sigma}{B}\sqrt{\frac{(\ln N+2)}{(K+\mathsf{nc})T}}$, we finish the proof of Lemma 3. □

Therefore, by combining the upper bound of the base regret with that of the meta regret, we finish the proof of the overall dynamic regret bound as stated in Theorem 1. □

C. Proof of Theorem 2

Proof of Theorem 2. We first show that the proposed sketching and ensemble pruning mechanisms can reduce the computational complexity of the algorithm. Note that we build an unbiased estimator in (3) by leveraging the offline dataset S_0 and current unlabeled dataset S_t. Besides, the online ensemble mechanism in HANOL needs a total of $\mathcal{O}(\log T)$ base learners to construct the ensemble model. Therefore, the computational complexity of the algorithm is $\mathcal{O}(|S_0| \cdot \log T)$ per round, which is costly for large-scale datasets. To address this, we propose the balanced kernel herding mechanism to sketch the offline dataset, store only mK samples, and also introduce the ordering-based ensemble pruning mechanism to update only N base learners. Therefore, the computational complexity is reduced to $\mathcal{O}(mKN)$ per round.

Then, we analyze the error introduced by the sketching and ensemble pruning mechanisms. Suppose the Reduced Kernel Hilbert Space $\mathcal{H}_k$ norm of the loss function is upper bounded by L, i.e., $L \triangleq \sup_{(\mathbf{x},y)\in\mathcal{X}\times\mathcal{Y},\mathbf{w}\in\mathcal{W}} \|\ell(f(\mathbf{w},\mathbf{x}),y)\|_{\mathcal{H}_k}$. We have $\|\widehat{R}_t(\mathbf{w}) - R_t(\mathbf{w})\| = \mathcal{O}(L/m)$ by the convergence rate of the kernel herding method [44], which finishes the proof. □

VI. Experiments

In this section, we present the empirical evaluations, which encompass experiments on five benchmark datasets and two real-world tasks related to the N-OLS problem. Our evaluation aims to answer the following questions:

- *Q1:* Does HANOL outperform other contenders in N-OLS when confronted with various types of shifts?
- *Q2:* Does HANOL show effectiveness in real-world tasks with the arrival of new classes and continuous label shift?
- *Q3:* Does each component of HANOL individually improve the performance? Does it correctly detect shifts and estimate the proportion of the new class? Is HANOL efficient?

A. Benchmark Datasets

This section seeks to answer **Q1**. We compare our proposed algorithm HANOL with seven competing methods using five benchmark datasets in the N-OLS scenario. Due to the novelty of the problem we are considering, there are currently no online algorithms specifically designed to address this problem. Therefore, the competing methods comprise a baseline approach (*FIX*), two for managing distribution shifts (*FTFWH* [4] and *ASL* [52]), two for handling the new classes in data streams (*SENC-F* [12] and *KNNENS* [38]), and two originally designed methods to tackle the offline N-OLS problem (*Self-N* and *PULSE* [53]). For the offline N-OLS methods, we made necessary modifications to adapt them to our specific setting. The details of all the competing methods are presented below.

- *FIX* is a baseline method that predicts with the initial classifier trained with offline data without online adaptation.
- *FTFWH* [4] is short for Follow The Fixed Window History, which averages across previously estimated priors within a sliding window. We set the window length as 100.
- *ASL* [52] is short for Augmented Self-Labeling method, which ensembles pseudo labels of different data augmentation-based models to handle distribution shifts.
- *SENC-F* [12] is short for SENC-Forest, a tree-based method to detect and classify the new class data.
- *KNNENS* [38] explores the local neighborhood information to handle the new class by employing an ensemble-based nearest neighbor technique.
- *Self-N* is a simple solution for the N-OLS problem where we directly combine the self-labeling and the new class

TABLE I

AVERAGE ERROR (%) OF DIFFERENT ALGORITHMS ON BENCHMARK DATASETS WITH DIFFERENT TYPES OF ENVIRONMENTAL SHIFTS, WHERE HANOL REPRESENTS OUR METHOD

				Gradual Shift				
	FIX	FTFWH [6]	ALS [52]	SENC-F [12]	KNNENS [38]	Self-N	PULSE [53]	HANOL
CIFAR10	22.89 ± 0.81 ○	19.01 ± 0.73 ○	19.23 ± 0.97 ○	18.92 ± 0.89 ○	19.23 ± 0.79 ○	19.11 ± 0.85 ○	18.71 ± 0.85	**18.52 ± 0.89**
CINIC10	35.57 ± 1.03 ○	30.08 ± 1.08 ○	31.45 ± 0.24 ○	30.25 ± 0.93 ○	30.12 ± 1.05 ○	31.95 ± 0.86 ○	29.89 ± 1.12 ○	**28.82 ± 0.96**
EuroSAT	16.23 ± 0.03 ○	10.82 ± 0.21 ○	11.24 ± 0.13 ○	10.55 ± 0.04 ○	10.91 ± 0.06 ○	11.13 ± 0.25 ○	**9.62 ± 0.16**	9.73 ± 0.06
Fashion	13.34 ± 0.13 ○	12.59 ± 0.16 ○	11.35 ± 0.23 ○	12.13 ± 0.51 ○	11.91 ± 0.32 ○	11.72 ± 0.05 ○	10.01 ± 0.09	**9.89 ± 0.02**
MNIST	4.98 ± 0.17 ○	3.12 ± 0.02 ○	2.56 ± 0.78	3.01 ± 0.09 ○	2.87 ± 0.17 ○	2.98 ± 0.05 ○	**2.43 ± 0.14**	2.56 ± 0.06
				Periodical Shift				
	FIX	FTFWH [6]	ALS [52]	SENC-F [12]	KNNENS [38]	Self-N	PULSE [53]	HANOL
CIFAR10	24.28 ± 0.72 ○	20.19 ± 0.82 ○	20.98 ± 0.79 ○	20.20 ± 0.77 ○	20.43 ± 0.77 ○	20.56 ± 0.81 ○	20.11 ± 0.81	**19.94 ± 0.74**
CINIC10	36.82 ± 1.03 ○	31.24 ± 0.91 ○	33.31 ± 0.52 ○	31.46 ± 1.15 ○	31.52 ± 1.15 ○	32.29 ± 0.88 ○	31.31 ± 1.12	**30.88 ± 1.04**
EuroSAT	17.72 ± 0.29 ○	12.12 ± 0.16 ○	11.89 ± 0.49 ○	11.72 ± 0.21 ○	12.22 ± 0.11 ○	12.61 ± 0.11 ○	10.84 ± 0.05 ○	**9.93 ± 0.17**
Fashion	14.75 ± 0.19 ○	14.04 ± 0.36 ○	12.96 ± 0.32 ○	13.56 ± 0.45 ○	13.27 ± 0.48 ○	12.67 ± 0.25 ○	**10.82 ± 0.12**	11.02 ± 0.44
MNIST	6.41 ± 0.15 ○	4.36 ± 0.16 ○	4.23 ± 0.15 ○	4.44 ± 0.02 ○	3.93 ± 0.09	4.02 ± 0.19 ○	3.82 ± 0.06	**3.75 ± 0.04**
				Sudden Shift				
	FIX	FTFWH [6]	ALS [52]	SENC-F [12]	KNNENS [38]	Self-N	PULSE [53]	HANOL
CIFAR10	23.58 ± 0.74 ○	19.24 ± 0.87 ○	19.56 ± 0.35 ○	19.23 ± 0.88	19.54 ± 0.82 ○	19.45 ± 0.76 ○	19.39 ± 0.88 ○	**18.88 ± 0.86**
CINIC10	36.21 ± 0.89 ○	33.33 ± 1.15	32.41 ± 0.72 ○	30.77 ± 1.12	30.64 ± 0.94	32.45 ± 0.94 ○	**30.55 ± 1.12**	31.26 ± 0.82
EuroSAT	16.79 ± 0.16 ○	11.15 ± 0.16	11.23 ± 0.45 ○	11.12 ± 0.07	11.55 ± 0.08 ○	11.36 ± 0.23 ○	10.18 ± 0.01	**10.06 ± 0.21**
Fashion	13.64 ± 0.24 ○	12.96 ± 0.37 ○	12.12 ± 0.07 ○	12.62 ± 0.01 ○	12.21 ± 0.37 ○	12.09 ± 0.05 ○	11.61 ± 0.26 ○	**10.92 ± 0.23**
MNIST	5.52 ± 0.05 ○	3.48 ± 0.13 ○	3.23 ± 0.23	3.45 ± 0.16 ○	3.44 ± 0.16 ○	3.24 ± 0.21	3.11 ± 0.13	**3.08 ± 0.09**

We report the mean and standard deviation over five runs. The best algorithms are emphasized in bold. "◦" indicates the algorithm is significantly inferior to our algorithms by paired t-test at a 5% significance level. The online sample size is set as $n_t = 10$.

detector. Self-N first initializes a model, then repeatedly minimizes empirical risks based on pseudo labels generated by the last classifier to handle distribution shifts. And it detects new classes by the tree-based method [12].

- *PULSE* [53] is a two-stage method that first estimates the fraction of the new class, then guides the classification of the target data. At each round, it retrains the model with the offline labeled data and the current unlabeled data, without reusing historical information.

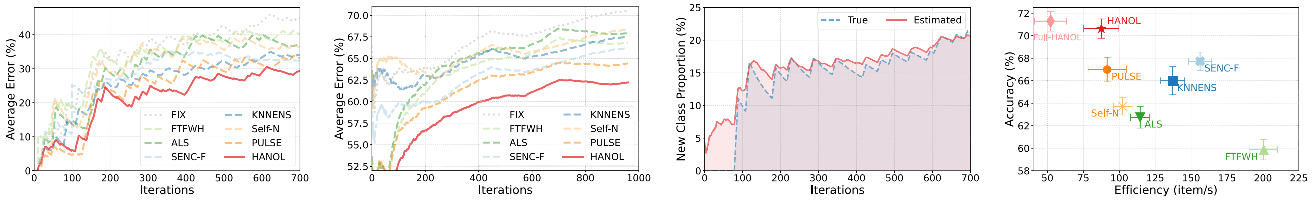
For the benchmark datasets, we generate a changing environment where the label distributions shift over time, and the new class data emerge in the online adaptation stage, which is not contained in the offline training data. In online adaptation stage, the learner can only observe unlabeled data streams. Specifically, we randomly choose two classes as the new classes for each benchmark dataset. The label distribution at round t is a mixture of two different constant distributions $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \Delta_{K+1}$ with a time-varying coefficient α_t, i.e., $\mathcal{D}_t(y) = (1-\alpha_t)\boldsymbol{\mu} + \alpha_t\boldsymbol{\mu}'$, where $\boldsymbol{\mu}_{y_t}$ denotes the distribution at round t and α_t controls the intensity of distribution changes. We only observe the label distribution $\boldsymbol{\mu}_0 \in \Delta_K$ for known classes in the offline training data. We simulate three typical types of distribution shifts in real-world tasks, specifically,

- `Gradual Shift`: the $\alpha_t = \frac{t}{T}$, which represents the gradual environmental change following a linear pattern.
- `Periodical Shift`: $\alpha_t = \sin\frac{i\pi}{L}$ periodically changes following a sinusoidal pattern, where $i = t \bmod L$ and L is a given periodic length. By default, we set $L = \Theta(\sqrt{T})$.
- `Sudden Shift`: At every iteration, we keep $\alpha_t = \alpha_{t-1}$ with a probability $p \in [0,1]$, otherwise set $\alpha_t = 1 - \alpha_{t-1}$. In the experiments, the parameter is set as $p = 1/\sqrt{T}$.

We evaluate all the contenders by average error over $T = 10,000$ rounds, with the following five benchmark datasets: CIFAR10, CINIC10, EuroSAT, Fashion, and MNIST.

Implementation Details: For the aforementioned five benchmark datasets, we employ a fine-tuned ResNet34 network for feature extraction. Images used to train the ResNet do not overlap with either the offline or online datasets. We sample 30, 000 data for offline initialization. We repeat all experiments for five times and evaluate the average error and standard deviation. The learning rates of the algorithms are set according to theoretical guidelines. The hyperparameter ε for the meta learner is set as $\sqrt{(\ln N)/T}$. δ and γ in MPE are set as the default values following [43], i.e., 0.1 and 0.01, respectively, without modification. The window size in the sliding-window MPE is set to $L = 20$ by default, without deliberate selection. Enhanced performance could be potentially achieved by selecting the window size using techniques such as cross-validation. In all experiments, we set the sketch size in our balanced kernel herding mechanism to 1, 000, and the number of base experts to 3 in our ordering-based pruning mechanism without careful tuning. All experiments are executed on a computer equipped with 2 Intel Xeon 8358 CPUs, each having 32 cores.

Results on Benchmark Datasets: The comparison results with the seven contenders on benchmark datasets are reported in Table I. These results demonstrate that our proposed algorithm effectively handles the new classes in the online label shift problem, outperforming other approaches. The baseline *FIX* is inferior to the online algorithms, highlighting the necessity of sequentially updated algorithms with online unlabeled data. Our method surpasses both *FTFWH* and *ASL*, indicating that handling the new class is crucial in the N-OLS setting. Besides,



(a) Accuracy curve on SHL (b) Accuracy curve on fMoW (c) New class estimation on SHL (d) Efficiency comparison

Fig. 3. (a) & (b) Comparison of overall performances on the real-world tasks. (c) Accuracy of the estimated new class proportion of our sliding-window MPE module. (d) Evaluation of efficiency and accuracy (defined as 100% - average error) of different algorithms. We report the mean and standard deviation over five runs. An algorithm closer to the top-right corner indicates superior efficiency and performance. *HANOL* is our approach. *Full-HANOL* is a variant of *HANOL* that is not equipped with the sketching and ensemble pruning mechanisms.

TABLE II
AVERAGE ERROR (%) OF DIFFERENT ALGORITHMS ON THE REAL-WORLD APPLICATIONS OF SHL [15] AND FMOW [16] DATASETS

	FIX	FTFWH	ALS	SENC-F	KNNENS	Self-N	PULSE	HANOL
SHL	44.82 ±1.12	40.14 ±1.35	37.26 ±1.42	32.28 ±1.24	34.01 ±1.87	36.28 ±1.15	33.01 ±1.65	**29.37** **±1.28**
fMoW	70.58 ±3.15	66.82 ±2.25	67.94 ±2.92	66.24 ±3.76	67.54 ±1.14	68.39 ±2.85	64.44 ±2.14	**62.22** **±3.03**

The performance metrics reported include both the mean accuracy and the standard deviation of different algorithms over a total of five separate runs. The best are emphasized in bold.

compared with *SENC-F* and *KNNENS*, which primarily focus on managing new classes, our method achieves better performance. This indicates that label shifts can lead to the misclassification of the new classes, and our black box shift estimator effectively tackles this issue. Our HANOL algorithm consistently outperforms both *PULSE* and *Self-N*, showing the effectiveness of our online updating scheme with sliding window-based MPE and online ensemble. These results show the success of our approach in tackling the N-OLS problem.

B. Real-World Applications

In this part, we aim to answer **Q2**. We compare the proposed approach with other contenders on two real-world applications: (i) the SHL locomotion recognition dataset [15], and (ii) the Functional Map of the World (fMoW) dataset [16], a sequential satellite image recognition task. The details of these applications are presented as below.

- SHL: This dataset is designed for human locomotion recognition using multi-modal sensor data (acceleration, gyroscope, gravity, pressure, etc.) collected from a body-worn camera and four smartphones at different body locations. We sample 30,000 offline and 77,000 online data points over an 11-day period, with six classes: still, walking, running, bike, car, and bus. During online updates, samples are processed chronologically based on timestamps, with label shifts and new classes emerging over time.
- fMoW: A satellite imagery dataset for building and land use classification, containing 83,412 images from over 200 countries with 63 building categories. Each image includes metadata about location, time, and environmental factors. The data stream is timestamp-ordered, with 10,000 earliest samples for offline initialization. Label distributions and building categories evolve between 2002 and 2017.

We report the average error of various algorithms on the real-world SHL and fMoW datasets in Table II, along with their respective timely performance depicted in Fig. 3(a) and (b). As shown in these empirical studies, our proposed method exhibits superior performance compared to the *FTFWH* and *ASL* methods, highlighting the significance of addressing the arrival of new classes in real-world tasks. Moreover, our proposed approach, HANOL, effectively adapts to label shift by the black box shift estimator and constructs a novel risk estimator for the N-OLS problem through the exploitation of unlabeled data, thereby outperforming the *SENC-F* and *KNNENS* methods. Our approach also surpasses the *PULSE* and *Self-N* methods, thanks to the benefits of the online updating scheme and the proposed sliding-window MPE mechanism, which alleviate the lack of labeled data problems in the online data streams.

C. Ablation Study

In this part, we aim to answer **Q3**. We conduct ablation studies of our proposed algorithm to validate the contribution of each component to the overall performance improvement. Additionally, we also report their running efficiency.

Modular Analysis: In order to demonstrate the benefits of the designed modules in HANOL, we quantitatively evaluate our proposed method and its variants by removing some components, i.e., (i) a baseline method that employs the risk estimator and stochastic gradient descent for model updating, but disregards the online ensemble structure and relies on only a single model; (ii) a variant of our proposal that does not utilize the sliding window-based MPE for handling the emergence of the new classes; and (iii) a method that excludes the black box shift estimator used to tackle the distribution shifts in the data streams. All the experiments are under the same hyperparameters for fair comparisons. The δ and γ in MPE are set as the default values for all experiments following [43], i.e., 0.1 and 0.01, respectively, without modification. The window size in the sliding-window MPE module is set to $L = 20$ by default, without deliberate selection. Enhanced performance could be potentially achieved by adaptively selecting the hyperparameters using techniques such as cross-validation.

As illustrated in Table III, employing the online ensemble structure significantly improves the performance in terms of accuracy, suggesting that an ensemble of multiple base learners can effectively handle the unknown distribution changes and the

TABLE III
ABLATION STUDY OF OUR PROPOSED HANOL ALGORITHM

ID	Ensemble	SW-MPE	BBSE	SHL	fMoW
(i)	-	√	√	37.54 ± 1.02	67.84 ± 2.87
(ii)	√	-	√	31.84 ± 1.15	65.37 ± 2.15
(iii)	√	√	-	32.67 ± 1.03	66.52 ± 2.87
HANOL	√	√	√	**29.37 ± 1.28**	**62.22 ± 3.03**

Ensemble represents the online ensemble structure, SW-MPE is the sliding-window MPE module to handle the arrival of new classes in the data stream, and BBSE represents the black-box estimator for the label shift estimation.

TABLE IV
HYPERPARAMETER SENSITIVITY ANALYSIS OF OUR ENSEMBLE PRUNING MECHANISM

Base Learner Num	$N = 1$	$N = 3$	$N = 5$	Full-HANOL
Error (%)	32.15 ± 2.87	29.37 ± 1.28	29.25 ± 1.19	28.62 ± 1.13
Efficiency (item/s)	92.39 ± 12.8	87.62 ± 12.3	69.34 ± 12.1	53.34 ± 10.6

We compare the error rate and efficiency (items processed per second) under different numbers of base learners N. Full-HANOL represents using all base models without pruning.

lack of labeled data problem in the data stream. Removing the sliding window-based MPE module causes a significant drop in the performance, thereby validating the effectiveness of the MPE module in managing the emergence of new classes. The performance further improves after employing the black box shift estimator, indicating that addressing the distribution shift is a critical aspect of the N-OLS problem, where label shift and the presence of new classes occur simultaneously.

Additionally, as demonstrated in Fig. 3(c), our proposed sliding-window MPE module is capable of accurately estimating the proportion of the emerging new classes, thereby managing the challenge of emerging new classes in the N-OLS problem effectively.

Efficiency Comparison: We also compare the efficiency of different algorithms. Specifically, we evaluate and compare the efficiency (items processed per second) and accuracy (defined as 100% - average error) of various algorithms. An algorithm that plots closer to the top-right corner indicates superior efficiency and performance since it achieves a better performance with higher efficiency. As demonstrated in Fig. 3(d), the moving average-based *FTFWH* is the most efficient, but it yields the poorest performance. Though the ensemble-based methods, *ALS* and *KNNENS*, exhibit slower speed, they accomplish superior performance. Our method, albeit with a slight compromise on efficiency, attains the best performance among all algorithms. Additionally, note that without our sketching and ensemble pruning mechanisms, although *Full-HANOL* achieves a slight performance improvement, it requires nearly twice the computational complexity. This is due to the need to store all offline labeled data and a much larger ensemble size compared to our HANOL, resulting in significantly slower processing speed.

Hyperparameter Sensitivity Analysis: We conduct a hyperparameter sensitivity analysis of our proposed algorithm to validate the sensitivity of hyperparameters. Specifically, we vary the number of base learners N to examine its effect on both performance and efficiency. As shown in Table IV, setting N to 3 achieves a good balance between performance and efficiency. Therefore, we set $N = 3$ as the default number of learners in our experiments.

VII. CONCLUSION

In this paper, we investigate the problem of handling emerging new classes in online label shift. We proposed a novel method, called HANOL, to tackle both online label shift and the emergence of the new classes in unlabeled data stream. Specifically, we first build a risk estimator for unlabeled data stream via risk rewriting and mixture proportion estimation to handle both the presence of emerging new class and the distribution shift. Then, we employ the paradigm of online ensemble to adapt to the unknown continuous label shift. Additionally, we also introduce the sketching and ensemble pruning mechanisms to improve the computational efficiency of the algorithm, making it more practical for real-world applications. The proposed method enjoys a theoretical guarantee of dynamic regret, affirming its effectiveness in adapting to changing distributions. We conduct experiments on five benchmark datasets and two real-world applications to validate the effectiveness of our HANOL. Notably, our proposed method exhibits significant improvements, achieving an average accuracy gain of 10% for the SHL dataset and 4% for the fMoW dataset compared to state-of-the-art contenders.

REFERENCES

[1] Z.-H. Zhou, *Machine Learning*. Singapore: Springer Nature, 2021.

[2] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. Cambridge, MA, USA: MIT Press, 2012.

[3] Z.-H. Zhou, "Open-environment machine learning," *Nat. Sci. Rev.*, vol. 9, no. 8, 2022, Art. no. nwac123.

[4] R. Wu, C. Guo, Y. Su, and K. Q. Weinberger, "Online adaptation to label distribution shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 11340–11351.

[5] J. Kang, A. Kang, A. Green, K. Gwee, and K. Ho, "Systematic review: Worldwide variation in the frequency of coeliac disease and changes over time," *Alimentary Pharmacol. Therapeutics*, vol. 38, no. 3, pp. 226–245, 2013.

[6] P. Zhao, J.-W. Shan, Y.-J. Zhang, and Z.-H. Zhou, "Exploratory machine learning with unknown unknowns," *Artif. Intell.*, vol. 327, 2024, Art. no. 104059.

[7] Y. Bai, Y.-J. Zhang, P. Zhao, M. Sugiyama, and Z.-H. Zhou, "Adapting to online label shift with provable guarantees," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 29960–29974.

[8] D. Baby, S. Garg, T. Yen, S. Balakrishnan, Z. C. Lipton, and Y. Wang, "Online label shift: Optimal dynamic regret meets practical algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 65703–65742.

[9] Y.-Y. Qian, P. Zhao, Y.-J. Zhang, M. Sugiyama, and Z.-H. Zhou, "Efficient non-stationary online learning by wavelets with applications to online distribution shift adaptation," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 41383–41415.

[10] R. Wu, S. Datta, Y. Su, D. Baby, Y. Wang, and K. Q. Weinberger, "Online feature updates improve online (generalized) label shift adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 106924–106954.

[11] Y.-Y. Qian, Y.-H. Wang, Z.-Y. Zhang, Y. Jiang, and Z.-H. Zhou, "Adapting to generalized online label shift by invariant representation learning," in *Proc. 31st ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2025, pp. 1161–1172.

[12] X. Mu, K. M. Ting, and Z. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1605–1618, Aug. 2017.

[13] X. Cai, P. Zhao, K. Ting, X. Mu, and Y. Jiang, "Nearest neighbor ensembles: An effective method for difficult problems in streaming classification with emerging new classes," in *Proc. 21th Int. Conf. Data Mining*, 2019, pp. 970–975.

[14] P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou, "Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization," *J. Mach. Learn. Res.*, vol. 25, no. 98, pp. 1–52, 2024.

[15] H. Gjoreski et al., "The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42592–42604, 2018.

[16] G. A. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in *Proc. 31st IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6172–6180.

[17] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, "Learning with feature and distribution evolvable streams," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 11317–11327.

[18] P. Zhao, X. Wang, S. Xie, L. Guo, and Z.-H. Zhou, "Distribution-free one-pass learning," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 951–963, Mar. 2021.

[19] S. Zhou et al., "Online continual adaptation with active self-training," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, 2022, pp. 8852–8883.

[20] H. Lian, D. Wu, B.-J. Hou, J. Wu, and Y. He, "Online learning from evolving feature spaces with deep variational models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 8, pp. 4144–4162, Aug. 2024.

[21] X. Luo, C. Liu, G. Gou, G. Xiong, Z. Li, and B. Fang, "Identifying malicious traffic under concept drift based on intraclass consistency enhanced variational autoencoder," *Sci. China Inf. Sci.*, vol. 67, no. 8, 2024, Art. no. 182302.

[22] J. Wang and X. Geng, "Explaining the better generalization of label distribution learning for classification," *Sci. China Inf. Sci.*, vol. 68, no. 5, 2025, Art. no. 152102.

[23] S.-Y. Li, S.-J. Zhao, Z.-T. Cao, S.-J. Huang, and S. Chen, "Robust domain adaptation with noisy and shifted label distribution," *Front. Comput. Sci.*, vol. 19, no. 3, 2025, Art. no. 193310.

[24] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognit. Lett.*, vol. 33, no. 2, pp. 191–198, 2012.

[25] A. Bifet, "Adaptive learning and mining for data streams and frequent patterns," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 55–56, 2009.

[26] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proc. 30th Very Large Data Bases*, 2004, pp. 180–191.

[27] P. Zhao, L. Zhang, Y. Jiang, and Z.-H. Zhou, "A simple approach for non-stationary linear bandits," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 746–755.

[28] C.-Y. Wei and H. Luo, "Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach," in *Proc. 34th Conf. Learn. Theory*, 2021, pp. 4300–4354.

[29] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 226–235.

[30] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *J. Mach. Learn. Res.*, vol. 8, pp. 2755–2790, 2007.

[31] P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou, "Dynamic regret of convex and smooth functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12510–12520.

[32] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure," *Neural Comput.*, vol. 14, no. 1, pp. 21–41, 2002.

[33] Z. C. Lipton, Y.-X. Wang, and A. J. Smola, "Detecting and correcting for label shift with black box predictors," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3128–3136.

[34] M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 859–874, Jun. 2011.

[35] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021.

[36] P. R. M. Júnior et al., "Nearest neighbors distance ratio open-set classifier," *Mach. Learn.*, vol. 106, no. 3, pp. 359–386, 2017.

[37] Y. Yu, W. Qu, N. Li, and Z. Guo, "Open category classification by adversarial sample generation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3357–3363.

[38] J. Zhang, T. Wang, W. W. Y. Ng, and W. Pedrycz, "KNNENS: A k-nearest neighbor ensemble-based method for incremental learning under data stream with emerging new classes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9520–9527, Nov. 2023.

[39] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th Int. Conf. Data Mining*, 2008, pp. 413–422.

[40] Y.-Y. Qian, Y. Bai, Z.-Y. Zhang, P. Zhao, and Z.-H. Zhou, "Handling new class in online label shift," in *Proc. 23rd IEEE Int. Conf. Data Mining*, 2023, pp. 1283–1288.

[41] H. G. Ramaswamy, C. Scott, and A. Tewari, "Mixture proportion estimation via kernel embeddings of distributions," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2052–2060.

[42] Y.-J. Zhang, P. Zhao, L. Ma, and Z.-H. Zhou, "An unbiased risk estimator for learning with augmented classes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 10247–10258.

[43] S. Garg, Y. Wu, A. J. Smola, S. Balakrishnan, and Z. C. Lipton, "Mixture proportion estimation and PU learning: A modern approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8532–8544.

[44] Y. Chen, M. Welling, and A. J. Smola, "Super-samples from kernel herding," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 109–116.

[45] X.-Z. Wu, W. Xu, S. Liu, and Z.-H. Zhou, "Model reuse with reduced kernel mean embedding specification," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 699–710, Jan. 2023.

[46] P. Zhao, Y.-F. Xie, L. Zhang, and Z.-H. Zhou, "Efficient methods for non-stationary online learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 11573–11585.

[47] G. Tsoumakas, I. Partalas, and I. P. Vlahavas, *An Ensemble Pruning Primer*, vol. 245. Berlin, Germany: Springer, 2009.

[48] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[49] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–936.

[50] M. Roberts, P. Mani, S. Garg, and Z. C. Lipton, "Unsupervised learning under latent label shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 18763–18778.

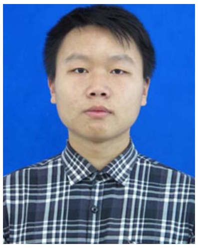
[51] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire, "Fast convergence of regularized learning in games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2989–2997.

[52] H. Yan, Y. Guo, and C. Yang, "Augmented self-labeling for source-free unsupervised domain adaptation," in *Proc. NeurIPS 2021 Workshop Distribution Shifts Connecting Methods Appl.*, 2021.

[53] S. Garg, S. Balakrishnan, and Z. C. Lipton, "Domain adaptation under open set label shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 22531–22546.

Yu-Yang Qian received the BSc degree from the University of Electronic Science and Technology of China in 2020. Currently, he is working toward the PhD degree with the School of Artificial Intelligence, Nanjing University. His research interest is mainly on machine learning and data mining.

Yong Bai received the BSc degree from Beihang University in 2020 and the MSc degree from Nanjing University in 2023. Currently, he is working with the Kuaishou Technology as a machine learning engineer. His research interest is mainly on machine learning and data mining.



Zhen-Yu Zhang received the BSc degree from the University of Electronic Science and Technology of China and the PhD degree from Nanjing University. He is currently a postdoctoral researcher with RIKEN Center for Advanced Intelligence Project. His research interests are mainly in machine learning and data mining.

Peng Zhao (Member, IEEE) received the BSc degree from Tongji University in 2016 and the PhD degree from Nanjing University in 2021. He is currently an assistant professor with the School of Artificial Intelligence, Nanjing University. His research interests lie in the foundations of machine learning, with a focus on online learning, learning theory, and optimization. He has published more than 50 papers in top-tier journals like *Journal of Machine Learning Research* and IEEE/ACM Trans, as well as premier conferences including ICML, NeurIPS, and COLT. He regularly serves as an area chair for ICML and NeurIPS.

Zhi-Hua Zhou (Fellow, IEEE) received the BSc, MSc, and PhD degrees (Hons.) in computer science from Nanjing University, Nanjing, China, in 1996, 1998, and 2000, respectively. He joined Nanjing University as an assistant professor, in 2001, where he is currently a professor and the vice president. He is also the founding director of the LAMDA Group. He has authored the books Ensemble Methods: Foundations and Algorithms, Evolutionary Learning: Advances in Theories and Algorithms, Machine Learning, and has published more than 200 papers in top-tier international journals or conference proceedings. He holds more than 30 patents. His research interests are mainly in artificial intelligence, machine learning, and data mining. He is a fellow of ACM, AAAI, AAAS, etc. He received various awards/honors, including the National Natural Science Award of China, the IEEE Computer Society Edward J. McCluskey Technical Achievement Award, and the CCF-ACM Artificial Intelligence Award. He founded Asian Conference on Machine Learning (ACML). He is the president of IJCAI Trustee, a series editor of Lecture Notes in Artificial Intelligence (Springer), an advisory board member of *AI Magazine*, the editor-in-chief of *Frontiers of Computer Science*, and the associate editor-in-chief of *Science China Information Sciences*.